
Indexation multi-niveau pour la recherche globale et partielle d'images par le contenu ¹

Geneviève Jomier* — Maude Manouvrier^{2, *} — Vincent Oria** — Marta Rukoz***

* Université Paris Dauphine - LAMSADE - Place du M^{al} De Lattre de Tassigny 75775 Paris Cedex 16, France - e-mail : {manouvri, jomier}@lamsade.dauphine.fr

** College of Computing Sciences New Jersey Institute of Technology - University Heights Newark, NJ, 07102 USA - e-mail : oria@homer.njit.edu

*** Universidad Central de Venezuela - CCPD - Escuela de Computación, Ciudad Universitaria - Av. Los Ilustres, Apt. 470002 - Los Chaguaramos 1041 Caracas, Venezuela - e-mail : mrukoz@strix.ciens.ucv.ve

RÉSUMÉ. Cet article présente une structure d'index pour la recherche d'images par le contenu, l'arbre QUIP (acronyme anglais pour *Quadtree-based Index for Image Retrieval and Image Pattern search*). Dans notre approche, chaque image de la base est représentée par un descripteur dit multi-niveau, qui stocke les descripteurs des quadrants de l'image, obtenus par une décomposition de l'image en arbre quaternaire. L'arbre QUIP permet de regrouper les images en clusters, en fonction de la similarité de leurs quadrants. Cette structure d'index permet non seulement des recherches globales d'images par le contenu, en appliquant un filtrage multi-niveau via l'arbre quaternaire, mais aussi des recherches d'images similaires par région.

ABSTRACT. This article presents a quadtree-based data structure for effective indexing of images. An image is represented by a multi-level feature vector, computed by a recursive decomposition of the image into four quadrants and stored as a full fixed-depth balanced quadtree. A node of the quadtree stores a feature vector of the corresponding image quadrant. A more general quadtree-based structure called QUIP-tree (*QUadtree-based Index for image retrieval and Pattern search*) is used to index the multi-level feature vectors of the images and their quadrants. A QUIP-tree node is an entry to a set of clusters that groups similar quadrants according to some pre-defined distances. The QUIP-tree allows a multi-level filtering in content-based image retrieval as well as partial queries on images.

MOTS-CLÉS : Similarité d'images et de régions d'images, arbre quaternaire, cluster, filtrage

KEYWORDS: Similarity search, similarity of image regions, quadtree, image cluster, filtering

1. Ce travail a été réalisé dans le cadre d'une coopération internationale CNRS - FONACIT/CDCH (accord 11996 et projet PI-03-13-5028-2002).

2. Auteur correspondant.

1. Introduction

Les systèmes de recherche d'images par le contenu (*Content-Based Image Retrieval systems* en anglais) [VEL 00] permettent de rechercher les images d'une base de données en fonction de leurs caractéristiques visuelles. Ces caractéristiques, encore appelées caractéristiques de bas-niveau, sont la couleur [SCH 01], la texture [PUZ 99] et la forme (ou contour) [CHA 00]. Elles sont stockées dans un vecteur numérique appelé *descripteur visuel*. En général dans ces systèmes, la requête est une image de la base de données, et le résultat de la requête correspond à une liste d'images ordonnées en fonction de la similarité entre leur descripteur visuel et celui de l'image requête, en utilisant une mesure de distance [RUI 99, SME 00]. Plus la distance entre les descripteurs visuels de deux images tend vers zéro, plus les images sont considérées comme similaires.

Dans plusieurs domaines d'application, l'utilisation de descripteurs résumant l'information photométrique globale des images, tels que les histogrammes de couleurs des images entières, n'offre pas toujours des résultats satisfaisants. Dans le domaine médical, par exemple, des descripteurs locaux, stockant les caractéristiques visuelles de régions d'image, sont nécessaires car le nombre de pixels représentant une pathologie est faible par rapport au nombre total de pixels dans l'image [SHY 98]. Une description globale ne permet pas de différencier suffisamment les images de la base et ne tient pas compte de la localisation des pixels et des régions d'intérêt. Pour remédier à cette limite et tenir compte de la localisation des caractéristiques visuelles dans le calcul de la similarité des images, plusieurs approches [AHM 03, ALB 00, GUP 97, LIN 01, LU 94, MAL 99, RUK 02] utilisent une structure spatiale, l'arbre quaternaire (en anglais *quadtree*) [SAM 84]. Une telle structure permet de stocker les caractéristiques visuelles des différentes régions d'image et de filtrer les images en augmentant au fur et à mesure le niveau de détails. Les images de la base sont tout d'abord comparées globalement avec l'image requête. Puis, si leur similarité globale avec l'image requête est inférieure à un certain seuil, les sous-régions d'images homologues sont comparées, ainsi de suite jusqu'aux régions de taille minimale. Cette technique d'indexation peut être utilisée pour comparer les images globalement [GUP 97, LU 94], mais peut également permettre de faire des requêtes sur les régions d'image telles que "*Retrouvez toutes les images de la base ayant la région nord-ouest et la région sud-est similaires à celles de l'image requête.*" [LIN 01, MAL 99].

La distance entre deux images représentées par des arbres quaternaires se calcule à partir des distances entre les quadrants homologues des deux images. Pour améliorer les performances des requêtes sur de telles images, plusieurs approches proposent une première étape de filtrage en utilisant un index sur les racines des arbres quaternaires (contenant les descripteurs des images entières), tel qu'un arbre R dans [LU 94], un arbre k-d dans [MAL 99] ou le hachage extensible dans [LIN 01]. Le filtrage, via l'index, diminue le nombre d'images devant être comparées région par région en utilisant l'arbre quaternaire, l'index regroupant les images en fonction de leur similarité globale. Après cette première étape de filtrage, une comparaison exhaustive des arbres quaternaires est faite pour les images appartenant au résultat du

filtrage. A notre connaissance, aucune approche ne propose de structure d'index permettant de regrouper les images en fonction, non plus seulement de leur similarité globale mais en fonction de la similarité de tous leurs quadrants. Cet article présente une telle structure, l'arbre *QUIP*. Cet arbre permet un filtrage multi-niveau des images non seulement pour de requêtes de similarité basées sur les caractéristiques globales des images mais aussi pour des requêtes portant sur la similarité locale des régions d'image. Cette structure indexe des images représentées par des arbres quaternaires en associant des mécanismes de regroupement (*clustering*) et de filtrage.

L'organisation de cet article est la suivante. La section 2 rappelle les concepts de base sur les descripteurs visuels, la représentation des images par des arbres quaternaires et le calcul de distances entre de telles images. La structure de l'arbre *QUIP*, que nous proposons, est définie dans la section 3. Des résultats d'expérimentation sont présentés dans la section 4. Finalement, la section 5 conclut cet article.

2. Concepts de base

2.1. Descripteurs visuels compacts ou résumés

Un descripteur visuel d'image est une représentation numérique des caractéristiques visuelles de l'image. Il existe des descripteurs de couleur, de forme ou de texture, pouvant être combinés. Pour chaque caractéristique, il existe différents modes de représentation. Pour plus de détails, le lecteur est invité à consulter [RUI 99, SME 00].

Les approches courantes de recherche d'images par le contenu [RUI 99] consistent à représenter le descripteur visuel de chaque image de la base par un point dans un espace multidimensionnel [BER 02], la similarité entre deux images étant proportionnelle à la proximité des points correspondants dans l'espace. Or, comme le précisent les auteurs de [CAS 03], les performances du calcul de distances se détériorent rapidement, proportionnellement au nombre d'images à comparer. A cet effet, des mécanismes d'indexation multidimensionnelle ont été proposés, comme par exemple l'arbre R [GUT 84]. Mais les performances de ce type d'index décroissent en fonction du nombre de dimensions [BEY 99]. Pour remédier à ce problème, il est possible de réduire les dimensions des descripteurs visuels. Dans le domaine de la couleur, par exemple, au lieu de représenter chaque image par un histogramme de 255^3 couleurs possibles¹, plusieurs approches telles que [FAL 94, LIN 01, VEN 01] réduisent la taille des descripteurs en calculant des statistiques sur la distribution des couleurs dans l'image. Les images sont, dans ce cas, représentées par des descripteurs dits compacts ou résumés. La figure 1 donne un exemple d'images représentées par des vecteurs moyenne des couleurs dans l'espace à trois dimensions $(R_{avg}, G_{avg}, B_{avg})$, dans lequel les moyennes des couleurs sont calculées pour les trois couleurs primaires rouge, vert et bleu (*Red, Green, Blue*). Pour plus de détails sur les vecteurs moyennes des

1. Une couleur peut être définie à partir des trois couleurs de base, rouge (R), vert (G pour *Green*), bleu (B); les valeurs d'une couleur de base pouvant varier de 0 à 255.

couleurs, il est possible de se référer à [FAL 94]. Les auteurs de [STR 95] proposent d'utiliser des descripteurs de couleur à 9 dimensions du type $(E_u, E_v, E_w, \sigma_u, \sigma_v, \sigma_w, s_u, s_v, s_w)$, basés sur les moments statistiques des couleurs : E représente la couleur moyenne, σ représente la variance et s , le moment d'ordre 3, pour chacune des dimensions de l'espace de couleur² (u, v, w) . D'autres approches utilisent des descripteurs de forme [CHA 00, KIM 00] ou des descripteurs de texture [PUZ 99].

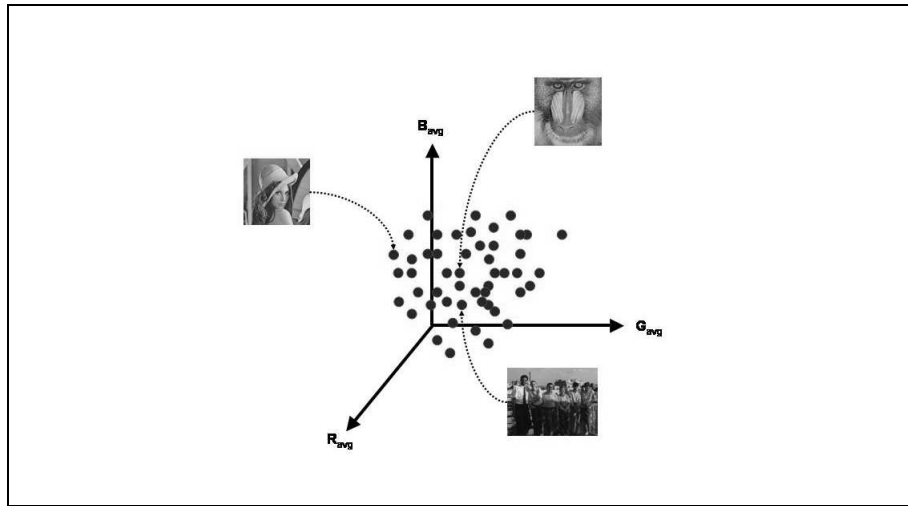


Figure 1. Un exemple d'images représentées dans l'espace $(R_{avg}, G_{avg}, B_{avg})$ des moyennes des couleurs.

La section suivante présente comment la combinaison de descripteurs visuels et de la structure d'arbre quaternaire peut être utilisée pour représenter le contenu des images à plusieurs niveaux de détails.

2.2. Descripteurs visuels multi-niveaux

L'arbre quaternaire (ou *quadtree*) [SAM 84] est une structure de données non équilibrée construite par division récursive de l'espace en quatre quadrants disjoints. L'arbre quaternaire est très utilisé dans le domaine des images, aussi bien pour le stockage [MAN 02], la compression, l'extraction d'information sur le contenu des images [ALB 00, CHA 00], que pour la recherche d'images par le contenu [KIM 00, LIN 01, LU 94, MAL 99, RUK 02]. Un état de l'art sur l'utilisation de l'arbre quaternaire dans le domaine des images est présenté dans [MAN 04].

Dans le domaine de la recherche d'images par le contenu, chaque image de la base de données peut être représentée par un arbre quaternaire équilibré complet dont

2. L'espace (u, v, w) peut être par exemple l'espace RGB, avec $u = R, v = G$ et $w = B$.

chaque nœud stocke un descripteur visuel du quadrant correspondant dans l'image. Ce descripteur peut être un histogramme de couleurs comme dans [GUP 97, LIN 01, LU 94], un descripteur de forme comme dans [KIM 00] ou un histogramme combinant à la fois la couleur et la texture comme dans [MAL 99]. Dans la suite de l'article, nous appellerons ce type de structure un *descripteur visuel multi-niveau*.

La figure 2.(a) présente l'image classique de *Lena* récursivement découpée en quatre quadrants pour être représentée par un descripteur multi-niveau. Les nœuds de l'arbre quaternaire et les quadrants correspondant dans l'image sont identifiés en suivant un ordre en Z, correspondant aux directions Nord-Ouest, Nord-Est, Sud-Ouest et Sud-Est. La racine de l'arbre et l'image entière sont identifiées par 0. Les entiers 0 à 3, précédés de 0, identifient les quatre premiers quadrants de l'image et les quatre nœuds d'arbre quaternaire associés. De manière récursive, les nœuds (resp. quadrants) fils d'un nœud (resp. sous-quadrants d'un quadrant), identifié par k , sont identifiés par kx avec x un entier prenant sa valeur dans $\{0, 1, 2, 3\}$.

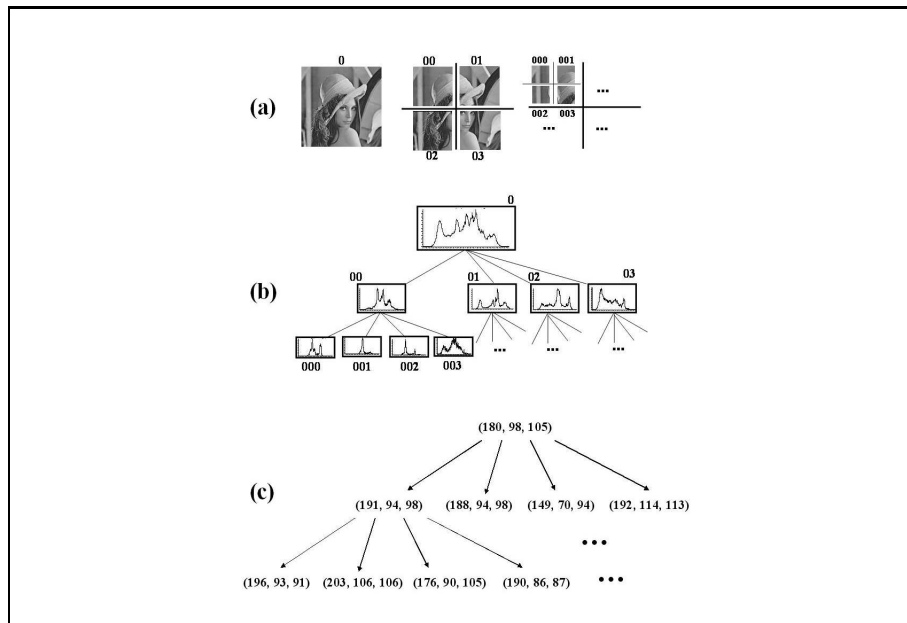


Figure 2. L'image de *Lena* : (a) décomposée en arbre quaternaire et représentée par (b) un descripteur de couleur multi-niveau basé sur les histogrammes de couleurs et (c) un descripteur de couleur multi-niveau basé sur la moyenne des couleurs.

Dans la suite de l'article, pour faciliter les explications et simplifier les figures, nous nous limiterons à des descripteurs de trois niveaux. La figure 2.(b) présente un exemple de descripteur de couleur multi-niveau de l'image de *Lena*, chaque nœud de l'arbre quaternaire contenant l'histogramme de couleurs de la région correspondante dans l'image. La figure 2.(c) représente un autre descripteur multi-niveau de l'image

de *Lena*, chaque quadrant de l'image étant représenté, dans ce cas, par son vecteur moyenne des couleurs dans l'espace à trois dimensions $(R_{avg}, G_{avg}, B_{avg})$.

Pour résumer, un descripteur multi-niveau d'image peut être représenté par un arbre quaternaire stockant les descripteurs de tous les quadrants de l'image. Chaque nœud de l'arbre quaternaire peut stocker n'importe quel type de descripteur, ce dernier pouvant être compact pour éviter les problèmes liés aux données de grandes dimensions. En outre, les descripteurs stockés dans les nœuds peuvent être différents en fonction du niveau des nœuds dans l'arbre quaternaire. Par exemple, la racine de l'arbre peut stocker un descripteur de couleur et ses quatre fils, représentant les quatre premiers quadrants de l'image, peuvent stocker des descripteurs de textures. Le choix des descripteurs de quadrants d'images stockés dans les nœuds d'arbre quaternaire dépend de l'application, des images de la base et des besoins des utilisateurs.

Un descripteur multi-niveau permet également de faire de la recherche d'images par similarité de régions. Une région r est, dans ce cas, composée de plusieurs quadrants d'image choisis par l'utilisateur dans une grille 4×4 ou 16×16 . L'utilisateur spécifie ensuite un vecteur de caractéristiques, puis un seuil pour chaque quadrant. Il peut également choisir les quadrants d'une image existant dans la base - voir Figure 3.(a) à (c). Le système détecte automatiquement le rectangle minimum englobant les quadrants sélectionnés par l'utilisateur - voir Figure 3.(d). La requête q_r ainsi déduite peut être transformées en plusieurs sous-requêtes, chaque sous-requête correspondant par exemple à une transformation géométrique (rotation ou translation) du rectangle minimum englobant. Dans [MAL 99], par exemple, la requête est décomposée en sous requêtes en effectuant des translations de la région initiale - la requête correspond à 6 sous-requêtes, celle de la Figure 3.(d) et celles de la Figure 3.(e).

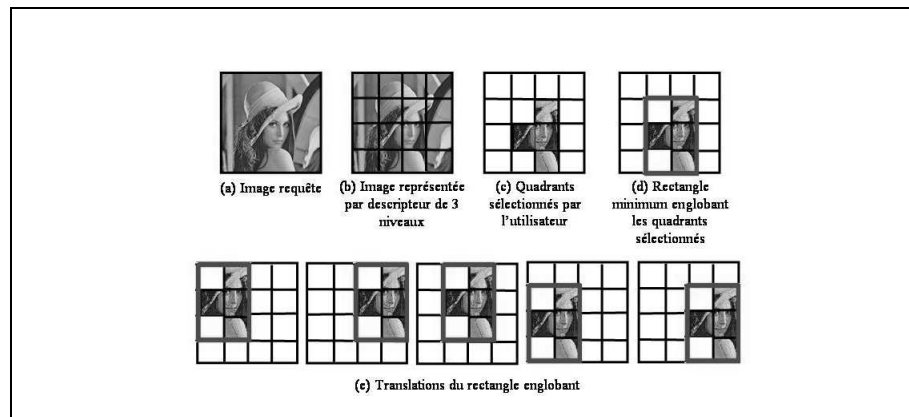


Figure 3. Un exemple de requête sur des régions d'image (adaptée de [MAL 99]).

2.3. Distances entre descripteurs multi-niveaux

Pour comparer deux images représentées par des descripteurs multi-niveaux, des distances entre arbres quaternaires ont été définies dans [RUK 02] et sont rappelées ci-après.

Dans la suite, on nomme *nœuds (resp. quadrants) homologues*, deux nœuds (resp. quadrants) de même identificateur, situés dans deux arbres quaternaires différents (resp. dans deux images différentes).

Soit $\delta_n(i, j)$, une distance métrique normalisée, $\delta_n(i, j) \in [0, 1]$, entre les deux descripteurs visuels des quadrants homologues n des images i et j . La distance δ_n peut être n'importe quelle distance de la famille de Minkowski³ et peut être pondérée.

Soit Δ , une distance entre deux images i et j , représentées par des descripteurs multi-niveaux. La distance Δ est définie comme la somme pondérée des distances normalisées δ_n entre les descripteurs, stockés dans chaque nœud homologue n des arbres quaternaires représentant les descripteurs multi-niveaux des images i et j :

$$\Delta(i, j) = \frac{\sum_n w_n \delta_n(i, j)}{\sum_n w_n} \quad [1]$$

Les coefficients w_n , $w_n \geq 0$, représentent les poids des distances δ_n dans le calcul de la distance Δ . La distance Δ est normalisée par le dénominateur $\sum_n w_n$ (au moins un des poids w_n doit être différent de zéro) : $\Delta \in [0, 1]$. Des cas particuliers de la distance Δ apparaissent dans [KIM 00, LU 94, LIN 01, MAL 99] - voir dans [MAN 04, RUK 02] pour plus de détails.

La valeur exacte de la distance Δ entre deux images représentées par des descripteurs multi-niveaux est obtenue par une comparaison exhaustive de tous les descripteurs des quadrants homologues. Cependant, cette valeur peut être approchée en comparant progressivement les descripteurs niveau par niveau, par un parcours en largeur d'abord des arbres quaternaires les représentant. Soit $\Delta^{(p)}(i, j)$, la distance jusqu'au niveau p de deux descripteurs multi-niveaux représentant les images i et j . Pour le calcul de la distance $\Delta^{(p)}(i, j)$, la condition suivante est imposée : $\delta_n(i, j) = 0$ pour tous les quadrants n dont la surface est inférieure à 4^{-p} par rapport à la surface de l'image entière (i.e. dont les nœuds correspondants dans le descripteur multi-niveau sont situés sous le niveau p , la racine étant au niveau 0). De manière intuitive, la distance $\Delta^{(p)}$ est calculée sans tenir compte des détails situés après un certain niveau p

3. Une distance de Minkowski, notée L_r , entre deux images i et j , représentées par des descripteurs (i_1, i_2, \dots, i_m) et (j_1, j_2, \dots, j_m) , est définie par $L_r(i, j) = \left[\sum_{d=1}^m |x[d] - y[d]|^r \right]^{\frac{1}{r}}$, avec r un entier, $r \geq 1$. Chaque dimension d peut être pondérée par un poids w_d non nul ($w_d |x[d] - y[d]|^r$), on parle dans ce cas de distance de Minkowski pondérée.

dans les arbres quaternaires représentant les descripteurs multi-niveaux⁴. Pour $p \geq 1$, il est évident que : $\Delta^{(p-1)}(i, j) \leq \Delta^{(p)}(i, j) \leq \Delta^{(p+1)}(i, j)$, la valeur de la distance Δ étant approchée niveau par niveau.

Dans certains cas, la prise en compte de tous les nœuds situés au-dessus du niveau p dans le calcul de la distance $\Delta^{(p)}$ peut gommer certaines informations issues du calcul de la distance. Prenons l'exemple simple de deux images représentant un damier noir et blanc (les quatre premiers quadrants étant complètement noirs ou complètement blancs), les quadrants noirs et blancs n'étant pas à la même position dans les deux images. La distance $\Delta^{(0)}$ (qui ne prend en compte que la racine des arbres quaternaires) entre les deux images est nulle puisque les deux images sont moitié noires moitié blanches. En revanche, pour le calcul de $\Delta^{(1)}$, si les poids w_n ont pour valeur $1/4$, chaque quadrant du niveau 1 occupant $1/4$ de l'image entière, on obtient une distance $\Delta^{(1)}$ égale à $1/2$. Dans ce cas, la valeur nulle de la distance au niveau zéro ($\Delta^{(0)}$) gomme le fait que la distance vaut 1 lorsque seuls les nœuds de niveau 1 sont pris en compte dans le calcul de la distance (les couleurs des quatre premiers quadrants étant différentes entre les deux images). Par conséquent, lorsque les descripteurs sont calculés à partir de méthodes statistiques, comme la moyenne des couleurs par exemple, ou lorsque le descripteur stocké dans un nœud père d'arbre quaternaire contient une approximation des descripteurs contenus dans ses nœuds fils, la distance Δ peut être calculée en ne prenant en compte que les nœuds de niveau p ($\forall n$, de niveau différent de p , $\delta_n(i, j) = 0$). L'inégalité $\Delta^{(p-1)}(i, j) \leq \Delta^{(p)}(i, j) \leq \Delta^{(p+1)}(i, j)$ est conservée, puisque les descripteurs des quadrants d'images, stockés au niveau p des arbres quaternaires, contiennent une approximation des descripteurs stockés au niveau $p + 1$ - voir dans [RUK 02] pour plus de détails.

Par résumer, le choix des poids w_n dépend fortement des caractéristiques choisies pour représenter les images et des métriques choisies pour les distances δ_n . Lorsque les descripteurs de quadrant sont la couleur dominante, comme dans [DEN 01], ou des descripteurs de forme tels que le descripteur du quadrant n n'est pas une approximation des descripteurs des sous-quadrants de n , comme dans [KIM 00], la distance $\Delta^{(p)}$ prend en compte tous les nœuds situés avant le niveau p et non pas uniquement les nœuds du niveau p . En revanche, lorsqu'un niveau du descripteur multi-niveau est une approximation du niveau inférieur, les nœuds de niveau p peuvent être pris en compte seuls dans le calcul de la distance $\Delta^{(p)}$.

La distance Δ peut être également utilisée pour évaluer la similarité entre régions d'image. Soit $\Delta_r(i, j)$, la distance Δ entre la région homologue r des images i et j , représentées par des descripteurs multi-niveaux. Cette distance est telle que $w_n \geq 0$ pour tous les quadrants n sélectionnés par l'utilisateur pour former la région r (voir à la fin de section 2.2) et telle que $w_n = 0$ pour les autres quadrants. Une image i possède une région similaire à la région requête q_r si la distance $\Delta_r(i, q_r)$ est inférieure à un certain seuil α .

4. Dans ce cas, $\Delta^{(p+1)} = \Delta^{(p)} + \frac{\sum_n w_n \delta_n(i, j)}{w}$ avec les nœuds n situés au niveau $p + 1$ et w la somme de tous les poids associés aux différents nœuds du descripteur multi-niveau.

3. Indexation multi-niveau

Cette section présente notre approche d'indexation multi-niveau pour la recherche globale et partielle d'images par le contenu. La structure que nous proposons est appelée arbre QUIP pour l'acronyme anglais pour *Quadtree-based Index for Image Retrieval and Image Pattern search*. Cette structure indexe les images représentées par des descripteurs multi-niveaux (voir Section 2.2). L'arbre QUIP est indépendant des descripteurs utilisés pour représenter les quadrants d'images. De même, n'importe quelle distance δ_n (voir Section 2.3) peut être utilisée pour comparer les quadrants homologues d'images. La seule contrainte est que le choix des descripteurs et des distances doit être fait de telle sorte que l'inégalité $\Delta^{(p-1)}(i, j) \leq \Delta^{(p)}(i, j) \leq \Delta^{(p+1)}(i, j)$ soit vérifiée. La section 3.1 présente le contexte. La section 3.2 définit la structure d'index que nous proposons et la section 3.3 présente les algorithmes de recherche d'images par le contenu.

3.1. Contexte

Les performances de l'exécution d'une requête de similarité d'images sont fortement dépendantes du nombre d'images à comparer, puisque l'algorithme de base consiste à calculer la distance de l'image requête avec toutes les images de la base. Pour améliorer ces performances, plusieurs approches ont été proposées, telles que les approches de regroupement (ou *clustering*) et les approches d'indexation.

Regrouper les images consiste à rassembler, au sein d'un même ensemble, des images ayant des caractéristiques similaires, la similarité étant calculée à l'aide d'une distance entre les descripteurs des images. Pour plus de détails, plusieurs états de l'art sur le regroupement de données sont présentés dans [HAL 01, JAI 99]. Parmi les algorithmes de regroupement, la méthode des nuées dynamiques (ou méthode des centres mobiles - en anglais *k-means*) [DID 71, MCQ 67] est la plus utilisée [HAL 01]. Il s'agit d'une approche facile à implémenter [JAI 99] et qui donne de bons résultats en terme de précision (*precision*) et rappel (*recall*) [LAU 98]. Cette approche consiste à répartir les données de la base, par exemple des images, en k groupes, répartis autour de k centres de groupes, appelés noyaux ou *centroïdes* : une image est dans un groupe (ou *cluster*) c si l'image est plus proche, en fonction d'une distance choisie, du centroïde du groupe c que de n'importe quel autre centroïde des autres groupes. Le centroïde de chaque groupe, recalculé à la fin de l'exécution de la méthode, correspond au barycentre de chaque groupe ou, dans le cas de l'approche des *k-medoids* [CHU 02], à une image du groupe.

Il existe également des techniques d'indexation multidimensionnelles pour gérer une base d'images. Un état de l'art est présenté dans [BER 02]. Le principe de ces structures est de découper l'espace et d'indexer les différentes régions de l'espace obtenues, à l'aide d'une structure arborescente, telle que l'arbre R [GUT 84]. L'objectif est de diminuer le nombre d'images à comparer et par conséquent le nombre de distances à calculer entre descripteurs, la structure d'index donnant directement accès à

un sous-ensemble des images de la base, ce sous-ensemble étant un sur-ensemble de l'ensemble résultat de la requête.

Il est possible de combiner une approche de regroupement avec une structure d'indexation. L'approche de [CAS 03] propose, par exemple, de partitionner l'ensemble des données à l'aide d'une méthode de regroupement, comme *k-means*, de diminuer les dimensions de l'espace vectoriel, en appliquant une décomposition en valeurs singulières sur chaque groupe ou *cluster*, puis de construire un index sur l'espace ainsi transformé.

La suite de cet article présente notre approche, combinant également une structure d'index, l'arbre quaternaire, et une méthode de regroupement (par ex. *k-means*), pour la recherche globale et partielle par le contenu d'images organisées en descripteurs multi-niveaux. Plusieurs approches telles que [AHM 03, ALB 00, GUP 97, LU 94, LIN 01, MAL 99] représentent les images par des arbres quaternaires pour faire de la recherche d'images par le contenu. Parmi ces approches, certaines [LIN 01, LU 94, MAL 99] utilisent une structure d'index au-dessus des arbres quaternaires représentant les images, dans le but de diminuer le nombre d'images, et donc d'arbres quaternaires, à comparer. Au lieu de faire un filtrage des images uniquement sur la racine des arbres quaternaires, en comparant les images globalement, nous proposons une structure d'index, l'arbre QUIP, regroupant les images en *clusters* en fonction de la similarité de tous leurs quadrants. Cette structure est présentée dans la section qui suit.

3.2. Structure de l'arbre QUIP

Dans notre approche, les images de la base sont représentées par des descripteurs multi-niveaux organisés en arbre quaternaire (voir section 2.2). Nous nous limiterons, dans la suite de l'article, à des descripteurs de trois niveaux. Néanmoins, notre approche peut se généraliser à des images représentées par des vecteurs stockés dans des arbres quaternaires de plus de trois niveaux.

Lorsqu'un descripteur multi-niveau est représenté par un arbre quaternaire complet de trois niveaux, chaque image de la base est représentée par 21 quadrants : le quadrant 0, représentant l'image entière, les quatre premiers quadrants Nord-Ouest, Nord-Est, Sud-Ouest et Sud-Est de l'image (identifiés par 00, 01, 02 et 03) et les 16 quadrants représentant les sous-quadrants des quadrants $0x$, $x \in \{0, 1, 2, 3\}$.

Chacun des 21 quadrants des images de la base est représenté par un descripteur dans un espace à plusieurs dimensions (par exemple à trois dimensions dans le cas des vecteurs moyennes de couleurs). Puis, chacun des 21 espaces à plusieurs dimensions est partitionné en utilisant une méthode de regroupement, comme celle des nuées dynamiques [DID 71, MCQ 67] par exemple. En utilisant l'approche des *k-means*, chaque groupe ou *cluster* est englobé par une hyper-sphère de centre μ et de rayon d , le rayon d définissant la distance entre le centre μ et l'image la plus éloignée du groupe.

Pour simplifier les explications et les illustrations ci-après, prenons l'exemple d'un espace de caractéristiques à trois dimensions, chaque quadrant d'image étant représenté par son vecteur moyenne des couleurs dans l'espace RGB (rouge/vert/bleu pour *Red/Green/Blue*). La partie droite de la figure 4 représente des descripteurs dans un espace à trois dimensions partitionné en quatre groupes, chaque groupe étant représenté par une sphère englobante. Les descripteurs des 21 quadrants d'une image sont stockés dans les nœuds de l'arbre quaternaire représentant le descripteur multi-niveau de l'image. Pour comparer les descripteurs stockés dans les descripteurs multi-niveaux, nous utilisons les distances présentées dans la section 2.3. Comme nous l'avons indiqué précédemment, notre approche est indépendante du choix des descripteurs de quadrants d'images et des distances utilisées entre ces descripteurs. La seule contrainte est que la distance au niveau p entre deux descripteurs multi-niveaux soit inférieure à leur distance au niveau $p + 1$.

Pour indexer ces 21 espaces multi-dimensionnels, nous proposons une structure, appelée l'arbre QUIP. L'arbre QUIP suit la décomposition des images : il s'agit d'un arbre quaternaire complet à 3 niveaux, lorsque les descripteurs multi-niveaux ont trois niveaux. Chaque nœud n de l'arbre QUIP contient des lignes. Une ligne l d'un nœud n de l'arbre QUIP associe un ensemble I_l d'identificateurs d'image à un groupe, ou *cluster*, c_l de quadrants d'image. Comme expliqué précédemment, un groupe c_l est représenté par son centre μ^{c_l} et son rayon d^{c_l} , le rayon d^{c_l} étant calculé à partir de la distance δ_n entre le descripteur du centre du groupe et celui du quadrant d'image n le plus éloigné dans le *cluster*. Par conséquent, chaque ligne l est un nuplet $[\mu^{(c_l)}, d^{(c_l)}, I_l]$. Lorsqu'un identificateur d'image i apparaît dans un ensemble I_l d'une ligne l de nœud n de l'arbre QUIP, cela signifie que le descripteur du quadrant n de l'image i appartient au groupe c_l : $\delta_n(\mu^{(c_l)}, i) \leq d^{(c_l)}$. Chaque nœud n de l'arbre QUIP est un ensemble de groupes ou *clusters* regroupant tous les quadrants n des images de la base.

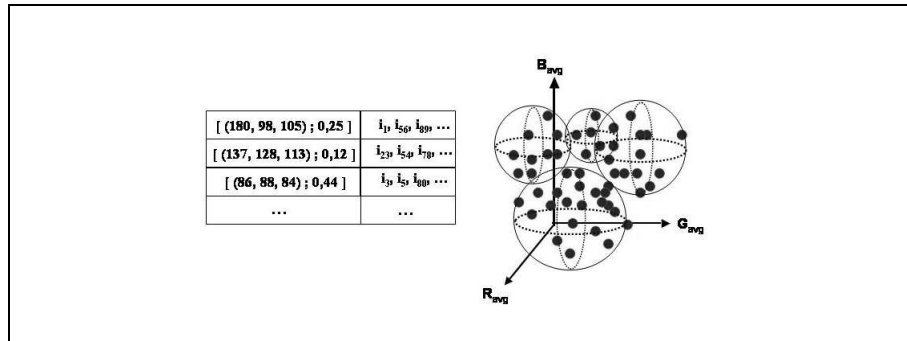


Figure 4. Un exemple de nœud de l'arbre QUIP.

La partie gauche de la figure 4 représente un nœud n d'arbre QUIP, les quadrants n des images de la base étant représentés par des vecteurs moyennes des couleurs regroupés en quatre *clusters*. La première ligne du nœud indique que le *cluster*, dont le centre a pour coordonnées (180, 98, 105) et pour rayon 0.25, regroupe les images

d'identificateurs i_1, i_{56}, i_{89} , etc. Les quadrants homologues n de ces images sont donc dans le même cluster et ont une distance δ_n inférieure à 0,25 avec le centre du cluster de coordonnées (180, 98, 105).

Lorsque le nombre de groupes, et donc le nombre de lignes par nœud de l'arbre QUIP, augmente, un index peut être construit pour chaque nœud afin d'accélérer l'accès aux groupes. Par exemple, si les nuées dynamiques [DID 71, MCQ 67] ont été utilisées pour former les groupes, un index de *clusters* peut être construit en appliquant la même méthode, comme c'est le cas par exemple pour le *Browsing Tree* de [BAR 04]. Il est également possible d'utiliser une structure d'index arborescente, comme l'arbre SR [KAT 97], chaque groupe stocké dans un nœud d'arbre QUIP correspondant, dans ce cas, aux feuilles de l'arbre SR indexant les groupes.

3.3. Algorithmes de recherche

L'arbre QUIP permet trois types de recherche d'images par le contenu : la recherche dite globale, la recherche de sous-images et la recherche partielle ou locale. La recherche globale consiste à retrouver des images similaires, selon la distance Δ , à une image requête en considérant les caractéristiques globales des images (donc tous les quadrants d'images depuis le quadrant 0). La recherche de sous-images consiste à comparer une image requête à tous les quadrants d'image stockés dans la base, quelles que soient la taille de l'image requête et celle des quadrants. La recherche locale ou partielle, quant à elle, consiste à retrouver des images ayant une région similaire, la région étant composée de quadrants d'image (voir section 2.2 et figure 3).

Dans la suite de cette section, nous nous limiterons aux requêtes par l'exemple où l'image requête est choisie parmi les images de la base. Il est, néanmoins, possible de généraliser l'approche. Nous ne tiendrons également compte que des requêtes du type à α -près (ou α -cut) : une image i est similaire à une image requête q lorsque $\Delta(i, q) \leq \alpha$, où α est un seuil donné et Δ est la distance entre arbres quaternaires définie dans la section 2.3. De même, dans cet article, $\Delta(i, q) = \Delta^{(2)}(i, q)$ puisque, afin de simplifier les explications, nous nous limitons dans nos exemples à des descripteurs de trois niveaux, la racine des arbres quaternaires étant située au niveau 0. Néanmoins, les algorithmes peuvent être généralisés à des descripteurs multi-niveaux et un arbre QUIP de plus de trois niveaux.

Le choix du seuil α dépend de la base d'images (il peut être fixé de manière empirique à partir de plusieurs tests sur la base) et des métriques choisies pour les distances δ_n . Le seuil peut également être adapté pour chaque niveau du descripteur multi-niveau. Soit α_p , le seuil au niveau p , i.e. le seuil de la distance $\Delta^{(p)}$. Par définition de la distance $\Delta^{(p)}$, le seuil α_p a une borne inférieure ou égale à $\Delta^{(p-1)}$ et une borne supérieure égale à $\Delta^{(p+1)}$. L'utilisateur peut donc choisir un seuil α_p pour chaque niveau p ou choisir un seuil α pour le dernier niveau (i.e. le seuil pour la distance $\Delta^{(2)}(i, q) = \Delta(i, q)$), ce dernier servant à calculer les seuils des niveaux supérieurs : $\alpha_0 \leq \dots \leq \alpha_p \leq \dots \leq \alpha$.

Recherche globale :

Pour exécuter une recherche globale, l'utilisateur doit choisir une image requête q , une distance δ_n (par défaut la distance euclidienne), un ensemble de poids w_n associés (par défaut $w_n = 4^{-p}$ où p est le niveau du nœud n dans le descripteur multi-niveau), ainsi qu'un seuil α . Le résultat de la requête est un ensemble d'images i ordonnées telles que $\Delta(i, q) \leq \alpha$, avec Δ la somme pondérée des distances δ_n entre les descripteurs des quadrants d'image (voir section 2.3). La recherche globale s'effectue en plusieurs étapes de filtrage :

Étape 1.(a) - Identification des groupes candidats : L'index des *clusters* (par exemple un *Browsing Tree* [BAR 04]) construit au-dessus du nœud 0 de l'arbre QUIP permet de déterminer les groupes, stockés dans le nœud 0, qui sont susceptibles de contenir des images pouvant appartenir à l'ensemble résultat, appelées par la suite *images candidates*. Les images candidates appartiennent à un *groupe candidat*. Un groupe c_l est candidat si $\Delta(\mu^{c_l}, q) - d^{c_l} \leq \alpha$, donc si la distance de l'image q au groupe c_l est inférieure au seuil α (i.e. si l'hyper-sphère de centre q et de rayon α coupe l'hyper-sphère de centre μ^{c_l} et de rayon d^{c_l}). Un groupe est candidat lorsqu'il contient des images candidates ou au moins un groupe candidat. Le parcours de l'index des *clusters* s'effectue comme suit. Le test $\Delta^{(0)}(\mu^{c_l}, q) - d^{c_l} \leq \alpha$ est appliqué depuis la racine de l'index des *clusters* à chaque groupe afin de reconnaître les groupes candidats (stockés dans les feuilles de la structure arborescente indexant les groupes).

Étape 1.(b) - Identification des images candidates : Les images candidates sont les images i appartenant aux groupes candidats, issus de l'étape 1.(a), et vérifiant $\Delta^{(0)}(i, q) \leq \alpha$. Pour déterminer ces images candidates, pour chaque image i des groupes candidats, on calcule la distance $\Delta^{(0)}(i, q)$. A la fin de cette première étape de filtrage, on obtient un ensemble \mathcal{R}_0 d'identificateurs d'image i telles que $\Delta^{(0)}(i, q) \leq \alpha$, i.e. dont la distance, par rapport à l'image requête, est inférieure à α , en ne tenant compte que du quadrant représentant l'image entière.

Pour chaque image i de l'ensemble \mathcal{R}_0 obtenu à l'étape précédente du filtrage, on effectue les étapes 2.(a) et 2.(b) successivement :

Étape 2.(a) - Calcul de la distance $\Delta^{(1)}(i, q)$: Ce calcul de distance est effectué en accédant aux nœuds du premier niveau (nœuds 00 à 03) du descripteur multi-niveau de l'image i . Cette étape élimine de \mathcal{R}_0 , les images i dont le quadrant 0 est similaire au quadrant 0 de l'image requête ($\Delta^{(0)}(i, q) \leq \alpha$), mais dont les sous-quadrants 00 à 03 ne sont pas similaires à ceux de l'image requête ($\Delta^{(1)}(i, q) > \alpha$).

Étape 2.(b) - Calcul de la distance $\Delta(i, q)$ lorsque $\Delta^{(1)}(i, q) \leq \alpha$: Ce calcul de distance est effectué en accédant aux nœuds feuille du descripteur multi-niveau de l'image i . Toutes les images i telles que $\Delta(i, q) \leq \alpha$, appartiennent à l'ensemble résultat \mathcal{R} .

Il est à noter que les calculs successifs des distances Δ utilisent les résultats obtenus à l'étape précédente. En effet, l'étape 2.(a) utilise le calcul de la distance $\Delta^{(0)}$, effectué à l'étape précédente 1.(b), et l'étape 2.(b) utilise le calcul de la distance $\Delta^{(1)}$,

effectué lors de l'étape précédente 2.(a). En résumé, la première étape de filtrage de la recherche globale réduit le nombre de comparaisons d'images en ne prenant en compte que les groupes d'images candidats. La deuxième étape filtre les images obtenues à l'issue de l'étape 1, en comparant uniquement les quatre premiers quadrants des images. Le résultat final est obtenu à l'issue de l'étape 2.(b) du filtrage. La comparaison exhaustive des descripteurs de tous les quadrants d'image ne s'effectue qu'à partir de l'étape 2.(a) et ceci pour un nombre restreint d'images (les images issues des groupes candidats et telles que $\Delta^{(1)}(i, q) \leq \alpha$).

Recherche de sous-images :

Dans notre approche, chaque image est représentée par un descripteur multi-niveau stockant les descripteurs de ses 21 quadrants, chaque quadrant pouvant être considéré comme une image à part entière. Par conséquent, une image requête q peut être comparée non seulement aux images entières, en appliquant une recherche globale, mais également en considérant chaque quadrant comme une image de la base. Si l'image requête q est comparée aux quadrants $0x$, $x \in \{0, 1, 2, 3\}$, deux niveaux de filtrage sont nécessaires. Dans ce cas, l'algorithme de recherche est analogue à celui d'une recherche globale exécutée à partir du nœud $0x$ de l'arbre QUIP. L'algorithme est répété pour chaque $x \in \{0, 1, 2, 3\}$. Si l'image requête q est comparée aux quadrants $0xy$, x et $y \in \{0, 1, 2, 3\}$, un seul niveau de filtrage est nécessaire, en recherchant uniquement les groupes candidats dans les nœuds $0xy$ de l'arbre QUIP. Cette recherche s'effectue 16 fois, autant que de nœuds $0xy$. Lorsqu'un quadrant d'image requête est comparé à toutes les images de la base, il est possible d'accéder au même niveau de finesse (i.e de considérer autant de détails pour le quadrant requête que pour les images de la base), dans la mesure où le quadrant requête n'est pas trop petit pour être découpé, en le décomposant par un arbre quaternaire ayant le même nombre de niveaux que les descripteurs multi-niveaux des images de la base.

Recherche locale ou partielle :

Lorsque l'utilisateur sélectionne une région requête q_r , composée de plusieurs quadrants (choisis par exemple dans une grille - voir section 2.2), une image i appartient à l'ensemble résultat si $\Delta_r(i, q_r) \leq \alpha'$. Le seuil $\alpha' = \sum_{||q_r||} \alpha$ où $||q_r||$ est le nombre de quadrants composant la région requête q_r . La recherche locale ou partielle suit un algorithme similaire à celui de la recherche globale, excepté qu'au lieu d'accéder à la racine de l'arbre QUIP, on accède aux nœuds n de l'arbre QUIP représentant les quadrants n qui composent la région requête q_r . Lorsque la région sélectionnée par l'utilisateur est composée de plusieurs quadrants d'image, la requête q_r est décomposée en sous-requêtes q_{r_n} , une sous-requête par quadrant n . L'algorithme s'applique donc à chaque quadrant n , créant ainsi plusieurs ensembles résultats \mathcal{R}_{r_n} d'identificateurs d'image. L'identificateur d'une image i apparaît dans l'ensemble résultat \mathcal{R}_{r_n} lorsque $\Delta_{r_n}(i, q_{r_n}) \leq \alpha$. Par conséquent, pour chaque image i dont l'identificateur apparaît dans l'union des ensembles \mathcal{R}_{r_n} , la distance $\Delta_r(i, q_r)$ est calculée, en accédant aux descripteurs de chaque quadrant n représentant la région r . L'ensemble résultat final \mathcal{R} contient les identificateurs des images i telles que $\Delta_r(i, q) \leq \alpha'$. Comme il est indi-

qué dans la section 2.2, il est possible d'appliquer des transformations géométriques, telles que la translation, à la région requête q_r .

La section qui suit présente quelques résultats d'expérimentation.

4. Résultats d'expérimentation

Un prototype implémentant les fonctionnalités de l'arbre QUIP a été développé en Java, au-dessus du SGBD Oracle 9.i. Les descripteurs multi-niveaux et les nœuds de l'arbre QUIP sont stockés dans la base de données de telle sorte que les requêtes de similarité puissent être exprimées facilement en SQL. Le calcul des distances est réalisé par des procédures externes à la base en Java.

Comme il est montré dans [LIN 01], lorsque des descripteurs de couleur sont basés sur des méthodes statistiques (moyenne, moment), il est inutile d'utiliser des descripteurs multi-niveaux de plus de trois niveaux : utiliser plus de trois niveaux augmente le nombre de comparaisons de quadrants mais ne change pas le résultat de la requête. Dans nos expérimentations, chaque image de la base est représentée par 21 descripteurs à 9 dimensions basés sur les moments statistiques des couleurs [STR 95], une image étant décrite par un arbre quaternaire de trois niveaux représenté sous la forme d'un vecteur de 189 valeurs. Nous avons pris des poids w_n proportionnels à la surface des quadrants d'image par rapport à l'image entière ($w_n = 4^{-p}$ avec p le niveau du nœud n dans l'arbre quaternaire associé à l'image). La distance δ_n choisie pour comparer les quadrants d'images est la distance euclidienne. Le regroupement des images est réalisé par l'algorithme *k-means* [DID 71, MCQ 67]. La base avec laquelle nous avons réalisé nos tests contient 2311 images de tableaux⁵ (peintures). Afin de valider les différentes étapes de filtrage, certaines images de la base ont été copiées et les copies ont subi des opérations de traitement d'images (distorsions, rotation, bruitage etc.). Chaque nœud de l'arbre QUIP contient 5 *clusters*⁶. Bien que nous ayons effectué des expérimentations pour tous les types de requête présentés dans la section 3.3, par manque de place, nous ne présentons dans cet article que les résultats obtenus pour la recherche globale utilisant l'arbre QUIP en comparaison avec une recherche séquentielle.

Comme nous l'avons expliqué précédemment (voir Section 3.3), chaque étape de la recherche globale diminue le nombre d'images à comparer. Le nombre d'images filtrées dépend du seuil choisi : plus le seuil est petit, meilleur est le filtrage. Sur la figure 5, par exemple, on peut voir qu'avec un seuil de 0,1, 32% des images de la base sont filtrées à l'étape 1.(a). Ces images sont éliminées par la recherche des groupes candidats (donc n'appartiennent pas aux groupes candidats), l'image requête étant comparées aux centres des groupes. Le taux de sélectivité est dans ce cas de 68% avec un seuil de 0,1 : 1575 images appartiennent aux groupes candidats. A l'étape

5. Images issues de <http://cgfa.sunsite.dk/fi/neart.htm>

6. Le nombre de 5 *clusters* par nœud a été choisi de manière expérimentale, afin que la répartition des images dans les différents *clusters* soit plus ou moins uniforme.

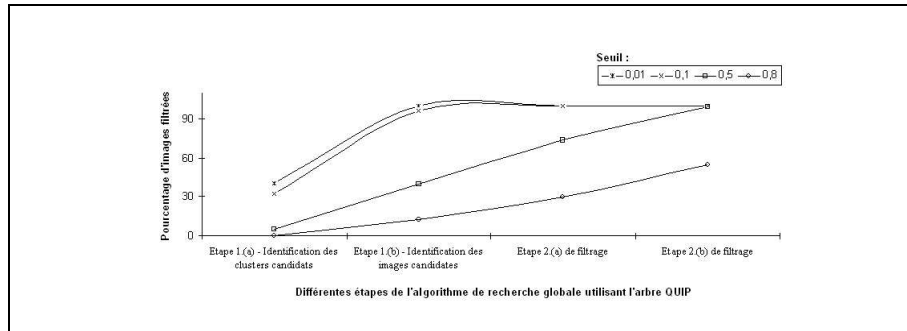


Figure 5. Le pourcentage d'images filtrées en fonction de différentes valeurs de seuils à chaque étape de la recherche globale.

suivante, avec le même seuil, le taux de sélectivité tombe à 4% : seulement 84 images sont sélectionnées à l'issue de l'étape 1.(b), après avoir calculé leur distance $\Delta^{(0)}$ avec l'image requête. Lorsque le seuil est de 0,5, le nombre d'images appartenant au résultat décroît, à chaque étape, de 2196 (à l'issue de l'étape 1.(a)), à 1397 (à l'issue de l'étape 1.(b)) et à 611 images (à l'issue de l'étape 2.(a)). Le résultat de la requête, obtenu à l'issue de l'étape 2.(b) du filtrage multi-niveau, pour un seuil de 0,5, contient 19 images. Avec un seuil encore plus grand (ex. 0,8), le taux de sélectivité est plus faible, mais le filtrage élimine néanmoins des images à chaque étape : l'étape 1.(a) retourne toute la base (2311 images - le seuil étant trop grand tous les groupes sont candidats), l'étape 1.(b) retourne 2021 images, l'étape 2.(a) en retourne 1622 et la dernière étape retourne 1049 images. L'étape 1.(b) est équivalente à une comparaison des images sur leurs caractéristiques globales (décrivant les images entières), puisque le calcul de la distance se fait uniquement entre les racines des descripteurs multi-niveaux. Par conséquent, une comparaison basée uniquement sur un descripteur global des images avec un seuil de 0,8 retournerait 2021 images, c'est-à-dire 87% de la base. En revanche, avec le même seuil, la recherche globale utilisant l'arbre QUIP a un taux de sélectivité meilleur : seulement 45% des images de la base sont considérées comme similaires à l'image requête (voir figure 5 pour le seuil 0,8). En effet, la recherche globale utilisant l'arbre QUIP prend en compte non seulement la similarité globale des images mais également celles de leurs quadrants homologues. Elle permet donc d'éliminer du résultat les images, qui bien que similaires globalement, ne sont pas similaires (en fonction de la distance Δ et du seuil α) au niveau de leurs quadrants homologues.

L'arbre QUIP est une structure efficace (i.e. plus rapide qu'une recherche séquentielle) lorsque le seuil α est petit. Une recherche séquentielle consiste à comparer, de manière exhaustive, les descripteurs multi-niveaux de toutes les images de la base avec le descripteur multi-niveau de l'image requête. Comme le montre la figure 6, une recherche globale utilisant l'arbre QUIP est 40% plus rapide qu'une recherche séquentielle, lorsque le seuil est faible (inférieur à 0,1). En revanche, avec un seuil supérieur

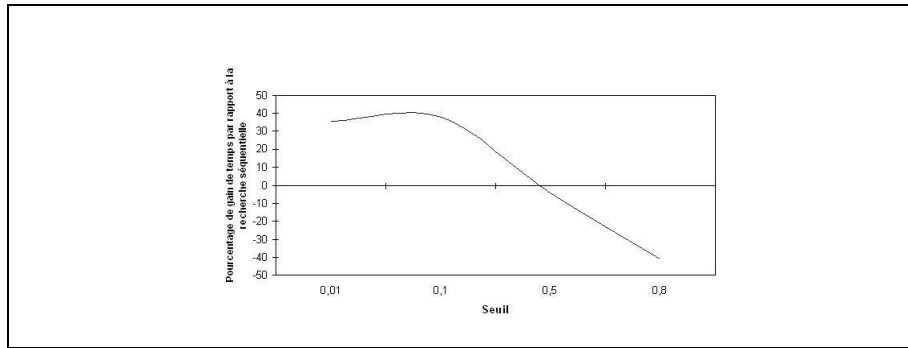


Figure 6. Gain ou perte de temps pour la recherche globale utilisant l'arbre QUIP par rapport à une recherche séquentielle, en fonction de différents seuils.

à 0,5, la recherche séquentielle est plus rapide. Il est à noter que le prototype actuel ne possède pas d'index de *clusters*. Par conséquent, pour le moment, l'image requête est comparée à tous les centres des *clusters* d'un nœud d'arbre QUIP.

La combinaison de l'arbre QUIP et des différentes étapes de filtrage améliore les performances de la requête, comme le montre la Figure 7, contenant les résultats obtenus pour 5 requêtes différentes⁷. En fonction de l'image requête, le gain de temps de la recherche globale en utilisant l'arbre QUIP (mais sans faire de filtrage par la suite) varie entre 5 et 21% par rapport à une recherche séquentielle. Lorsque le filtrage multi-niveau est effectué après l'utilisation de l'arbre QUIP (i.e. après la recherche des *clusters* candidats), le gain de temps est supérieur à 50% par rapport à une recherche séquentielle.

En termes de filtrage et de temps d'exécution des requêtes, l'arbre QUIP donne des résultats prometteurs.

5. Conclusion et discussion

Dans cet article, nous avons proposé une structure d'indexation d'images, un arbre quaternaire dont chaque nœud stocke le descripteur du quadrant correspondant dans l'image. Chaque image de la base est représentée par un descripteur multi-niveau et l'arbre QUIP permet d'organiser les descripteurs des images en fonction de leur similarité de quadrant. L'arbre QUIP combine une approche de regroupement et une structure d'arbre quaternaire, permettant ainsi de faire un filtrage multi-niveau et réduisant, par conséquent, le nombre d'images à comparer. L'originalité de cette structure vient du fait qu'elle peut supporter plusieurs types de requêtes : la recherche globale qui compare les images sur leurs caractéristiques globales, la recherche de sous-image qui

7. Pour une même image requête, nous avons exécuté plusieurs fois la même recherche, afin de calculer un temps d'exécution moyen pour une image requête donnée.

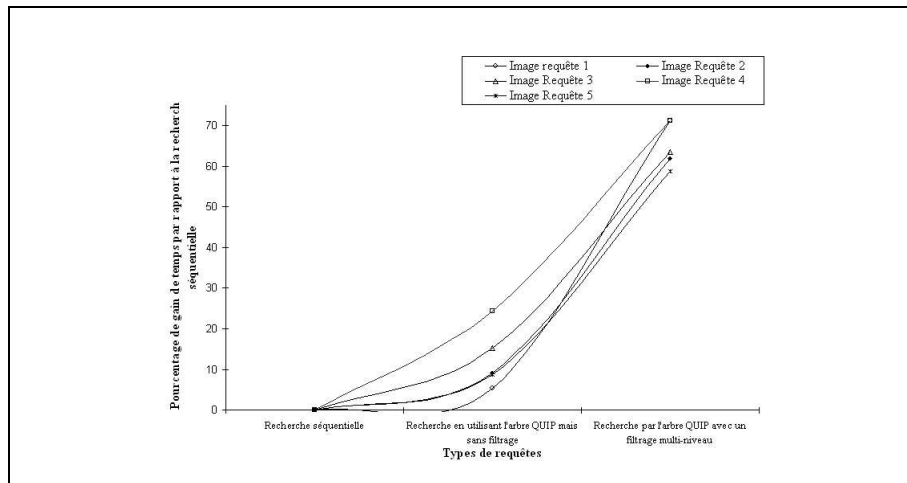


Figure 7. Gain de temps d'une recherche globale utilisant l'arbre QUIP par rapport à une recherche séquentielle, avec ou sans les différentes étapes de filtrage.

permet de comparer une image requête à tous les quadrants des images stockées dans la base ou un quadrant d'image requête à des images de la base, et la recherche locale ou partielle qui permet de retrouver des images ayant une région similaire.

Notre approche a pour objectif de diminuer le nombre de distances à calculer lors d'une requête de similarité. Les nœuds de l'arbre QUIP contenant des groupes ou *clusters* de quadrants d'image, les performances de la structure dépendent fortement du nombre de groupes. Lorsque le nombre de groupes devient critique, l'utilisation d'une structure pour indexer le contenu des nœuds de l'arbre QUIP est à envisager. Nous effectuons actuellement des expérimentations, afin de valider entièrement la structure QUIP. En particulier nous comptons évaluer le nombre critique de groupes par nœud. Nous allons également "passer à l'échelle" en travaillant sur des bases de données beaucoup plus volumineuses en terme d'images stockées.

Remerciements : Les auteurs de l'article tiennent vivement à remercier Maximilien Beau, Jean-Batiste Labrousse, Rosita E. Perozol et José Miguel Lanna, pour la réalisation du prototype et des expérimentations.

6. Bibliographie

- [AHM 03] AHMAD I., GROSKY W., « Indexing and retrieval of images by spatial constraints », *Journal of Visual Com. and Image Rep.*, vol. 14, n° 3, 2003, p. 291–320.
- [ALB 00] ALBUZ E., KOCALAR E., KHOKHAR A., « Quantized CIE Lab* Space and Encoded Spatial Structure for Scalable Indexing of Large Color Image Archives », *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, Istanbul (Turkey), June 2000.

- [BAR 04] BARTOLINI I., CIACCIA P., PATELLA M., « The PIBE Personalizable Image Browsing Engine », *1st Int. Workshop on Computer Vision meets Databases (CVDB)*, Paris (France), June 2004, p. 43–50.
- [BER 02] BERRANI S.-A., AMSALEG L., GROS P., « Recherche par similarité dans les bases de données multidimensionnelles : panorama des techniques d'indexation », *RSTI - Ing. des sys. d'information. Bases de données et multimédia*, vol. 7, n° 5-6, 2002, p. 9–44.
- [BEY 99] BEYER K., GOLDSTEIN J., RAMAKRISHNAN R., SHAFT U., « When Is "Nearest Neighbor" Meaningful ? », *7th Int. Conf. on Database Theory (ICDT'99)*, Sydney (Australia), 1999.
- [CAS 03] CASTELLI V., THOMASIAN A., LI C.-S., « CSVD : Clustering and Singular Value Decomposition for Approximate Similarity Search in High Dimensional Spaces », *IEEE Trans. on Knowledge and Data Eng. (TKDE)*, vol. 15, n° 3, 2003, p. 671–685.
- [CHA 00] CHAKRABARTI K., ORTEGA-BINDERBERGER M., PORKAEW K., ZUO P., MEHROTRA S., « Similar Shape Retrieval in MARS », *IEEE Int. Conf. on Multimedia and Expo (ICME II)*, New York, NY, USA, 2000, p. 709–712.
- [CHU 02] CHU S.-C., RODDICK J., PAN J., « An Efficient K-medoids-based Algorithm Using Previous Medoid Index, Triangular Inequality Elimination Criteria and Partial Distance Search », *4th Int. Conf. DaWaK*, Aix-en-Provence (France), 2002.
- [DEN 01] DE NATALE F., GRANELLI F., « Structured-Based Image Retrieval Using a Structured Color Descriptor », *Int. Workshop on Content-Based Multimedia Indexing (CBMI'01)*, Brescia (Italy), 2001.
- [DID 71] DIDAY E., « Une nouvelle méthode en classification automatique et reconnaissance des formes : la méthode des nuées dynamiques », *Revue de Statistique Appliquée*, vol. XIX, n° 2, 1971, p. 19–33.
- [FAL 94] FALOUTSOS C., EQUITZ W., FLICKNER M., NIBLACK W., PETKOVIC D., BARBER R., « Efficient and Effective Querying by Image Content », *Journal of Intelligent Information Systems*, vol. 3, n° 3/4, 1994, p. 231–262.
- [GUP 97] GUPTA A., JAIN R., « Visual Information Retrieval », *Communications of ACM (CACM)*, vol. 40, n° 5, 1997, p. 70–79.
- [GUT 84] GUTTMAN A., « R-trees : A Dynamic Index Structure for Spatial Searching », *ACM SIGMOD Int. Conf. on the Management of Data*, 1984, p. 45–57.
- [HAL 01] HALKIDI M., BATISTAKIS Y., VAZIRGIANNIS M., « On Clustering Validation Techniques », *Journal of Intelligent Information Sys.*, vol. 17, n° 2-3, 2001, p. 107–145.
- [JAI 99] JAIN A., MURTY M., FLYNN P., « Data clustering : a review », *ACM Computing Surveys*, vol. 31, n° 3, 1999, p. 264–323.
- [KAT 97] KATAYAMA N., SATOH S., « The SR-tree : An Index Structure for High-Dimensional Nearest Neighbor Queries », *ACM SIGMOD Int. Conf. on Management of Data*, Tucson, Arizona (USA), 1997, p. 369–380.
- [KIM 00] KIM H.-K., KIM J.-D., « Region-based shape descriptor invariant to rotation, scale and translation », *Signal Processing : Image Com.*, vol. 16, n° 1-2, 2000, p. 1–293.
- [LAU 98] LAU T. K., KING I., « Performance analysis of clustering algorithms for information retrieval in image databases », *Int. Joint Conf. on Neural Networks (IJCNN'98)*, 1998.
- [LIN 01] LIN S., TAMER ÖZSU M., ORIA V., NG R., « An Extendible Hash for Multi-Precision Similarity Querying of Image Databases », *27th Int. Conf. on Very Large Data-*

Bases (VLDB'2001), Roma (Italy), 2001.

- [LU 94] LU H., OOI B.-C., TAN K.-L., « Efficient Image Retrieval by Color Contents », *1st Int. Conf. on Applications of Databases (ADB-94)*, Vadstena (Sweden), 1994.
- [MAL 99] MALKI J., BOUJEMAA N., NASTAR C., WINTER A., « Region Queries without Segmentation for Image Retrieval by Content », *3rd Int. Conf. on Visual Information Systems (Visual'99)*, Amsterdam (The Netherlands), 1999.
- [MAN 02] MANOUVRIER M., RUKOZ M., JOMIER G., « Quadtree representations for storage and manipulation of clusters of images », *Image and Vision Computing*, vol. 20, n° 7, 2002, p. 513–527.
- [MAN 04] MANOUVRIER M., RUKOZ M., JOMIER G., « *Spatial Databases : Technologies, Techniques and Trend* », chapitre Quadtree-Based Image Representation and Retrieval, Y. Manolopoulos et al. (Eds), IGP, InfoSIC and IRM Press Pub., 2004, A paraître.
- [MCQ 67] MCQUEEN J., « Some methods for classification and analysis of multivariate observations », *5th Berkeley Symp. on Math., Statistics and Probability*, 1967, p. 281–298.
- [PUZ 99] PUZICHA J., RUBNER Y., TOMASI C., BUHMANN J., « Empirical Evaluation of Dissimilarity Measures for Color and Texture », *IEEE Int. Conf. on Computer Vision (ICCV'99)*, 1999, p. 1165–1173.
- [RUI 99] RUI Y., HUANG T., CHANG S.-F., « Image Retrieval : Current Techniques, Promising Directions and Open Issues », *Journal of Visual Com. and Image Rep.*, vol. 10, 1999, p. 39–62.
- [RUK 02] RUKOZ M., MANOUVRIER M., JOMIER G., « Distances de similarité d'images basées sur les arbres quaternaires », *18èmes Journées Bases de Données Avancées (BDA'2002)*, Evry (France), Oct. 2002.
- [SAM 84] SAMET H., « The Quadtree and Related Hierarchical Structures », *ACM Computing Surveys*, vol. 16, n° 2, 1984, p. 187–260.
- [SCH 01] SCHETTINI R., CIOCCA G., ZUFFI S., « *Color Imaging Science : Exploiting Digital Media* », chapitre "A Survey on methods for colour image indexing and retrieval in image databases", J. Wiley, 2001.
- [SHY 98] SHYU C., BRODLEY C., KAK A., KOSAKA A., AISEN A., BRODERICK L., « Local versus Global Features for Content-Based Image Retrieval », *IEEE Workshop on Content-Based Access of Image and Video Libraries (CBAIVL'98)*, Santa Barbara, California, 1998.
- [SME 00] SMEULDERS A., WORRING M., SANTINI S., GUPTA A., JAIN R., « Content based image retrieval at the end of the early years », *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, n° 12, 2000, p. 1349–1380.
- [STR 95] STRICKER M., ORENGO M., « Similarity of Color Images », *Storage and Retrieval for Image and Video Databases (SPIE)*, San Diego (USA), 1995.
- [VEL 00] VELTKAMP R., TANASE M., « Content-Based Image Retrieval Systems : A Survey », rapport n° UU-CS-2000-34, 2000, Dep. of Computing Science, Utrecht University.
- [VEN 01] VENEAU E., PÉPY J., VINET L., « Toward a semantic indexing based on images color features similarity ? », *Int. Conf. on Image and Signal Processing (ICISP'2001)*, Agadir (Morocco), 2001.